



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A Geometric Nonlinear AMLI Preconditioner for the Bingham Fluid Flow in Mixed Variables

A. Aposporidis, P. S. Vassilevski, A. Veneziani

November 9, 2012

Electronic Transactions on Numerical Analysis

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

A GEOMETRIC NONLINEAR AMLI PRECONDITIONER FOR THE BINGHAM FLUID FLOW IN MIXED VARIABLES *

ALEXIS APOSPORIDIS [‡] PANAYOT S. VASSILEVSKI [‡] AND ALESSANDRO VENEZIANI [†]

Abstract. In [3] we have introduced an augmented formulation for the Bingham fluid flow problem. The resulting solver based on this formulation has been proven to be more effective than the one in primitive variables with respect to the regularization parameter which is usually introduced for the numerical solution of the equations. In addition it converges even in absence of regularization without a significant degradation of the performance. However, the discretized augmented problem requires the solution of large and sparse linear systems with a twofold saddle point structure. For large scale applications, the set up of an efficient and robust preconditioner is mandatory. In this paper we suggest using the regularized Bingham problem as a preconditioner for the non-regularized problem. For effectively solving the regularized linear system, we introduce a geometric multigrid approach that involves recursive calls to coarse levels within a flexible GMRES method inspired by the nonlinear AMLI preconditioners considered in [33]. Numerical experiments demonstrate the effectiveness of this preconditioning technique.

AMS subject classifications. 65F10, 65N30, 65N55

Key words. multigrid; multilevel flexible GMRES; Bingham flow; mixed finite elements

1. Introduction. Many fluids of industrial, geophysical and medical interest exhibit a shear-dependent viscosity. In particular, visco-plastic materials show properties of a rigid continuum as long as the applied stress remains below a certain threshold and become incompressible fluids if this critical value is exceeded [9]. A common example of a visco-plastic material is the Bingham fluid [30]. If \mathbf{u} denotes the velocity field of an incompressible fluid in the domain Ω , p is the pressure, we denote by $D\mathbf{u} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ the strain rate tensor and consider the Frobenius norm $|D\mathbf{u}| = \sqrt{\text{tr}(D\mathbf{u}^T D\mathbf{u})}$. In Bingham fluids, for

$$(1.1) \quad \boldsymbol{\tau} = 2\mu D\mathbf{u} + \tau_s \frac{D\mathbf{u}}{|D\mathbf{u}|},$$

we solve the system

$$(1.2) \quad \begin{cases} \rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] - \nabla \cdot \boldsymbol{\tau} + \nabla p = \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

when $|\boldsymbol{\tau}| > \tau_s$. Here, $\mu > 0$ (*plastic viscosity*), $\rho > 0$ (*fluid density*) and $\tau_s \geq 0$ (*yield stress*) are assumed to be constant. When $|\boldsymbol{\tau}| \leq \tau_s$, we set

$$D\mathbf{u} = \mathbf{0}.$$

The region of Ω where the latter equation holds is called *rigid* or *plug* region, as opposed to the *fluid* region, where (1.2) is assumed to hold. The constitutive relation reads therefore

$$(1.3) \quad D\mathbf{u} = \begin{cases} \mathbf{0} & \text{if } |\boldsymbol{\tau}| \leq \tau_s \text{ (plug region),} \\ \left(1 - \frac{\tau_s}{|\boldsymbol{\tau}|}\right) \frac{\boldsymbol{\tau}}{2\mu} & \text{if } |\boldsymbol{\tau}| > \tau_s \text{ (fluid region).} \end{cases}$$

*THIS WORK WAS PERFORMED UNDER THE AUSPICES OF THE U.S. DEPARTMENT OF ENERGY BY LAWRENCE LIVERMORE NATIONAL LABORATORY UNDER CONTRACT DE-AC52-07NA27344.

[†]DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, EMORY UNIVERSITY, ATLANTA, GA

[‡]CENTER FOR APPLIED SCIENTIFIC COMPUTING, LAWRENCE LIVERMORE NATIONAL LABORATORY, 7000 EAST AVENUE, MAIL STOP L-560, LIVERMORE, CA 94550.

Equations (1.1), (1.2) can be viewed as a generalization of the Navier-Stokes equations with shear-dependent viscosity $\hat{\mu} = 2\mu + \frac{\tau_s}{|D\mathbf{u}|}$ in the fluid region, reducing to the classical Navier-Stokes equations with constant viscosity if $\tau_s = 0$. A major difficulty associated with solving the Bingham equations is that the flow and plug regions are unknown *a priori*. Notice that $\hat{\mu}$ is singular in the plug region where $|D\mathbf{u}|$ vanishes. These difficulties can be addressed by *regularizing* $\hat{\mu}$. The most common types of regularization are the Bercovier-Engelmann regularization [8], in which $|D\mathbf{u}|$ is replaced by $|D\mathbf{u}|_\varepsilon = \sqrt{|D\mathbf{u}|^2 + \varepsilon^2}$, and the Papanastasiou variant [26]. In practice, regularization techniques replace the plug region by a high viscosity flow region. Other methods are based on different formulations of the problem, such as the method introduced by Duvaut and Lions [17, 18]. This solver is based on a variational inequality and Uzawa-like iterative methods. In this paper, we consider the mixed formulation of the Bingham fluid flow introduced in [3]. An auxiliary symmetric tensor $W = \frac{D\mathbf{u}}{|D\mathbf{u}|}$ is defined and (1.2)-(1.3) are reformulated into

$$(1.4) \quad \begin{cases} \rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] - \nabla \cdot (2\mu D\mathbf{u} + \tau_s W) + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \\ D\mathbf{u} - |D\mathbf{u}|W = \mathbf{0} \end{cases}$$

Note that in this formulation contains no division by $|D\mathbf{u}|$ and hence no singularity occurs. In this respect, (1.4) is more regular than the primitive formulation (1.1), (1.2). The idea of circumventing a singularity by adding an unknown was inspired by [10], where a similar approach has been successfully applied to a total-variation based image processing problem.

A regularized formulation is obtained when $|D\mathbf{u}|$ is replaced by $|D\mathbf{u}|_\varepsilon = \sqrt{|D\mathbf{u}|^2 + \varepsilon^2}$. The steady Stokes-type equations for the Bingham flow is obtained when the Lagrangian or material derivative in square brackets in the momentum equation (1.4)₁ (or (1.2)₁) is dropped.

Results in [3] indicate that the iterative solution of the mixed formulation converges within a small number of iterations and is robust with respect to both mesh size and ε . In addition, the method efficiently solves the non-regularized Bingham problem ($\varepsilon = 0$). However, the price of the mixed formulation is that additional unknowns augment the system, resulting in larger linear systems to be solved at each nonlinear iteration. Numerical experiments in [3] are performed on two dimensional test problems and use a direct method for solving the linear systems. Direct methods are in general infeasible for large scale problems. The purpose of the present paper is to introduce an efficient and robust preconditioner for solving the mixed formulation for large problems, that are also suitable in three dimensions. We propose a preconditioning procedure which is based on two steps:

- (i) The *regularized* Bingham problem is used as a preconditioner for solving the *non-regularized* problem. In this respect, the Bercovier-Engelman regularization parameter ε serves as a control parameter driving the performance of the preconditioner rather than as a perturbation of the problem.
- (ii) The regularized problem is then approximately solved using a multilevel technique. In particular, we introduce a geometric multilevel preconditioner for which the smoothing is performed by a flexible **GMRES** (**FGMRES**) scheme preconditioned by an overlapping additive Schwarz domain decomposition method. The multigrid iterations are based on recursive cycles performed again within a flexible **FGMRES** scheme on different grids. The overall scheme gives rise to a nonlinear method sometimes referred to as nonlinear AMLI. We use a direct solver on the coarsest level.

Numerical results exhibit robustness and scalability of the linear solver with respect to the mesh size, demonstrating that the proposed method can be used for the accurate simulation of (non-regularized) Bingham fluids in real-life cases.

The paper is organized as follows. In Section 2 the problem setting, including the linearization and discretization of the problem, are given. The first part of Section 3 introduces the preconditioner based on the regularized Bingham problem. In the second part we describe the multilevel algorithm used to solve the regularized problem. Numerical results on two benchmark problems in two and three dimensions for several values of the mesh size, as well as a test case on a more complex geometry are presented in Section 4. Conclusions are drawn in Section 5.

2. Problem setting. We denote by $H^s(\Omega)$ the Sobolev space of functions with s distributional derivatives with summable square (H_0^1 denotes the set of H^1 functions with null trace on the boundary). In addition, $L^r(0, T; H^s)$ denotes the vector space of functions whose H^s norm for the spatial dependence is r -power summable in the time interval $(0, T)$. We use \mathbf{H}_0^1 for vector functions with components in H_0^1 and \mathcal{L}^2 for tensor functions with components in L^2 . If we assume for simplicity that the boundary conditions prescribe $\mathbf{u} = 0$ on $\partial\Omega$ (for $t > 0$), the weak formulation of (1.4) reads: for $\mathbf{f} \in L^2(0, T; L^2(\Omega))$, find $\mathbf{u} \in L^2(0, T; \mathbf{H}_0^1(\Omega))$, $p \in L^2(0, T; L^2(\Omega))$, $W \in L^2(0, T; \mathcal{L}^\infty(\Omega))$ s.t.

$$(2.1) \quad \left\{ \begin{array}{l} \rho \int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \mathbf{v} + \rho \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \mathbf{v} + \mu \int_{\Omega} D\mathbf{u} \mathbf{v} - \int_{\Omega} p \nabla \cdot \mathbf{v} + \tau_s \int_{\Omega} \nabla \cdot W \mathbf{v} = \int_{\Omega} \mathbf{f} \mathbf{v} \\ - \int_{\Omega} q \nabla \cdot \mathbf{u} = 0 \\ \int_{\Omega} Z : \nabla \mathbf{u} - \int_{\Omega} (|D\mathbf{u}|^2 + \varepsilon^2)^{1/2} W : Z = 0 \end{array} \right.$$

with $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$ a given initial condition in $\mathbf{L}^2(\Omega)$, for all $\mathbf{v} \in \mathbf{H}_0^1(\Omega)$, $q \in L^2(\Omega)$, and $Z \in \mathcal{L}^2(\Omega)$. For the time discretization, we refer to a classical Backward Euler method. Different, more accurate time advancing schemes can be considered as well. For the space discretization, we resort to finite elements. Again, different discretization techniques may be considered, such as finite difference schemes on staggered grids [22, 25] and finite volume discretizations [29]. In [3] we have proven that for $\varepsilon > 0$ no *inf-sup* constraint needs to be fulfilled in the selection of finite dimensional space of W for the well posedness of the discrete problem. The existence of an *inf-sup* constraint if $\varepsilon = 0$ is still an open problem.

The discrete Picard linearized associated problem reads: find $\mathbf{u}_h \in L^2(0, T; \mathbf{V}_h)$, $p_h \in L^2(0, T; Q_h)$. ■

$W_h \in L^2(0, T; \mathcal{Z}_h)$ such that

$$(2.2) \quad \left\{ \begin{array}{l} \frac{1}{\Delta t} \rho \int_{\Omega} \mathbf{u}_h^{n+1,k} \mathbf{v}_h + \rho \int_{\Omega} (\mathbf{u}_h^{n+1,k-1} \cdot \nabla) \mathbf{u}_h^{n+1,k} \mathbf{v}_h \\ \quad + \mu \int_{\Omega} D \mathbf{u}_h^{n+1,k} \mathbf{v}_h - \int_{\Omega} p_h^{n+1,k} \nabla \cdot \mathbf{v}_h \\ \quad + \tau_s \int_{\Omega} W_h^{n+1,k} : \nabla \mathbf{v}_h = \frac{1}{\Delta t} \rho \int_{\Omega} \mathbf{u}_h^{n,k} \mathbf{v} + \int_{\Omega} \mathbf{f}^{n+1} \mathbf{v}_h \\ \\ - \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h^{n+1,k} + \alpha \int_{\Omega} p^{n+1,k} q_h = 0 \\ \\ \int_{\Omega} Z_h : \nabla \mathbf{u}_h^{n+1,k} - \int_{\Omega} (|D \mathbf{u}_h^{n+1,k-1}|^2 + \varepsilon^2)^{1/2} W_h^{n+1,k} : Z_h = 0 \end{array} \right.$$

for all $\mathbf{v}_h \in \mathbf{V}_h$, $q_h \in Q_h$, and $Z_h \in \mathcal{Z}_h$. Here $n, n+1$ refer to the time step, Δt is the time step size, $k, k-1$ refer to the Picard iteration. The index h indicates the size of the space discretization mesh. Notice the pressure stabilizing term introduced in the mass conservation equation. The parameter α will be taken as small as 10^{-10} . The matrix formulation of the problem reads (we drop the time index for easiness of notation) $\mathcal{A}_{\varepsilon} \mathbf{w} = \mathbf{b}$ with

$$(2.3) \quad \mathcal{A}_{\varepsilon}(\mathbf{u}^{(k-1)}) = \begin{bmatrix} A(\mathbf{u}^{(k-1)}) & B^T & C^T \\ B & -\alpha Q & 0 \\ C & 0 & -N_{\varepsilon}(\mathbf{u}^{(k-1)}) \end{bmatrix},$$

$$\mathbf{w} = \mathbf{w}^{(k)} = \begin{bmatrix} \mathbf{u}^{(k)} \\ p^{(k)} \\ W^{(k)} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{f} \\ 0 \\ 0 \end{bmatrix}$$

for $k = 1, 2, \dots$ until convergence. The block $N_{\varepsilon}(\mathbf{u}^{(k-1)})$ is obtained by discretizing $|D \mathbf{u}^{(k-1)}|_{\varepsilon} W$. This block is symmetric positive definite provided that $\varepsilon > 0$.

The efficient solution of this system with complex geometries or a large number of degrees of freedom can be obtained based either on an approximate factorization of $\mathcal{A}_{\varepsilon}$, resorting to a sequential computation of velocity, pressure and the tensor W , or utilizing an efficient preconditioner. In the latter case, we could take advantage of the twofold saddle point-structure of the problem. As a matter of fact, notice that there are two different ways to recognize the saddle-point structure of (2.3). Letting

$$\mathcal{B} = \begin{bmatrix} B \\ C \end{bmatrix} \quad \text{and} \quad \mathcal{N}_{\varepsilon} = \begin{bmatrix} -\alpha Q & 0 \\ 0 & -N_{\varepsilon}(\mathbf{u}^{(k-1)}) \end{bmatrix}$$

gives a saddle-problem of the form

$$\mathcal{A}_{\varepsilon} = \begin{bmatrix} A & \mathcal{B}^T \\ \mathcal{B} & \mathcal{N}_{\varepsilon} \end{bmatrix}$$

with a positive definite (1,1)-block, which is also symmetric in the case of the Stokes-type problem. On the other hand, one may define

$$\mathcal{F} = \begin{bmatrix} A & B^T \\ B & -\alpha Q \end{bmatrix} \quad \text{and} \quad \mathcal{G} = \begin{bmatrix} C & 0 \end{bmatrix}.$$

In this case, the problem becomes

$$\begin{bmatrix} \mathcal{F} & \mathcal{G}^T \\ \mathcal{G} & -N_\varepsilon(\mathbf{u}^{k-1}) \end{bmatrix}$$

and the (1,1)-block of the saddle-point problem is indefinite and represents in turn a saddle-point problem. Many preconditioners have been suggested for saddle-point problems either when the matrix (1,1)-block of the system is s.p.d (symmetric positive definite), or its symmetric part is s.p.d. A broad spectrum of preconditioners relies on inexact factorizations of the system and an approximation of the Schur complement, such as the least square commutator preconditioner or the pressure convection diffusion preconditioner [19, 20]. Other preconditioning techniques for saddle-point problems include augmented Lagrangian preconditioners [6, 7] or preconditioners based on a dimensional splitting [4, 5]. Here, we propose a preconditioner for \mathcal{A}_0 (the non-regularized problem) based on a multilevel monolithic approximation of \mathcal{A}_ε to achieve a method with potentially optimal complexity.

3. The multilevel preconditioner.

3.1. Approximating the non-regularized problem. As we have pointed out in the introduction, one of the main advantages of the mixed formulation from [3] is the treatment of the singularity represented by the plug regions. The numerical solver based on this formulation is more robust with respect to the regularization parameter. Numerical results show that the mixed formulation can be used for the non-regularized case. The idea we pursue here is to regard *the regularized mixed formulation of the Bingham problem as a preconditioner for the non-regularized case*. In other terms, we use a preconditioner built up for the regularized problem for solving the case $\varepsilon = 0$. To support this idea, in Figure 3.1, we report the eigenvalues of the non-regularized Bingham matrix \mathcal{A} for the case of the Stokes-type equations computed for one of our test cases, the flow between parallel plates (see Section 4.3) with a number of degrees of freedom small enough to use MATLAB's `eig`, namely $h = 1/16$ in a 2D unit square. Figure 3.1 also displays the eigenvalues of \mathcal{A} when preconditioned by the regularized problem \mathcal{A}_ε , i.e. the eigenvalues of $\mathcal{A}_\varepsilon^{-1}\mathcal{A}$ with $\varepsilon = 10^{-2}$. Clustering of the eigenvalues around $\lambda = 1$ is evident.

To quantify the impact of the regularization parameter on the non-regularized problem, we define

$$\mathcal{S} = \begin{bmatrix} A & B^T \\ B & -\alpha Q \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} B \\ C \end{bmatrix}$$

and consider the following factorization of the linear system matrix \mathcal{A}_ε with $\varepsilon > 0$:

$$\mathcal{A}_\varepsilon = \begin{bmatrix} \mathcal{S} & \mathcal{F}^T \\ \mathcal{F} & N_\varepsilon \end{bmatrix} = \begin{bmatrix} \mathcal{S} & 0 \\ \mathcal{F} & N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T \end{bmatrix} \begin{bmatrix} I & \mathcal{S}^{-1}\mathcal{F}^T \\ 0 & I \end{bmatrix}.$$

Then

$$\mathcal{A}_\varepsilon^{-1} = \begin{bmatrix} I & -\mathcal{S}^{-1}\mathcal{F}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{S}^{-1} & 0 \\ (N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T)^{-1}\mathcal{F}\mathcal{S}^{-1} & -(N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T)^{-1} \end{bmatrix}$$

and by a direct computation we get

$$(3.1) \quad \mathcal{A}_\varepsilon^{-1}\mathcal{A} = \begin{bmatrix} I & -\mathcal{S}^{-1}\mathcal{F}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \mathcal{S}^{-1}\mathcal{F}^T \\ 0 & -\mathcal{X} \end{bmatrix}$$

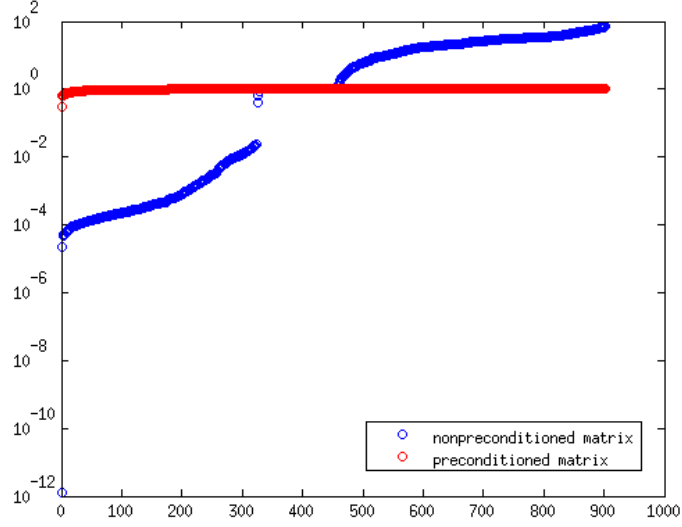


FIG. 3.1. Absolute values of the eigenvalues of the discrete linearized Bingham matrix \mathcal{A} (blue) and eigenvalues of $\mathcal{A}_\varepsilon^{-1}\mathcal{A}$ (red) in the analytical test case, where \mathcal{A}_ε is the regularized Bingham matrix with $\varepsilon = 10^{-2}$.

with the matrix block

$$(3.2) \quad \mathcal{X} = (\mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T) - N_\varepsilon^{-1}(\mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T - N).$$

Note that \mathcal{S} represents the (Newtonian) Stokes (or Navier-Stokes) part of the linear system and the inverse \mathcal{S}^{-1} is well-defined provided either \mathbf{u} and p are discretized in *inf-sup* compatible spaces or $\alpha > 0$. We can see from (3.1) and (3.2) that the eigenvalues of the preconditioned matrix $\mathcal{A}_\varepsilon^{-1}\mathcal{A}$ cluster around one if the spectrum of $(N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T)$ is similar to the spectrum of $(\mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T - N)$. Should the inverses be computed exactly, this trivially holds true for $\varepsilon \rightarrow 0$. Figure 3.2 shows the residual for the first 30 iterations of **GMRES** when solving the preconditioned system $\mathcal{A}_\varepsilon^{-1}\mathcal{A}\mathbf{x} = \mathcal{A}_\varepsilon^{-1}\mathbf{f}$ for different values of ε . To provide this proof of concept, a coarse grid is used (again unit square domain with $h = 1/16$) and the inverse of \mathcal{A}_ε is applied exactly using a direct method. The smaller ε , the faster the **GMRES** iterations reach any given tolerance.

Note that so far in the literature on solving the Bingham fluid flow equations there has been a strict distinction between solvers for the regularized model and solvers for the non-regularized model. To the authors' knowledge, this type of combination of regularized and non-regularized model presented here has not been advocated before.

3.2. Approximating the regularized problem. Using the exact inverse of \mathcal{A}_ε as a preconditioner as done in Figure 3.1 is clearly not practical if the problem is large. The efficient solution of the linear system requires an approximation of the inverse which can be computed with a relatively low cost in terms of memory and CPU time and which significantly reduces the number of iterations of the linear iterative solver. In this paper we use a geometric multigrid technique for approximating $\mathcal{A}_\varepsilon^{-1}$.

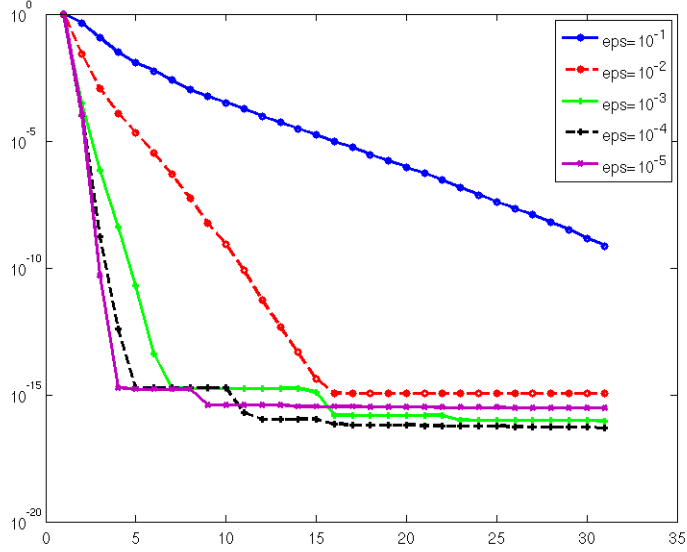


FIG. 3.2. *Residual of GMRES for the first 30 iterations for when solving the non-regularized problem preconditioned by the regularized one with different values of ε .*

Multigrid (MG) methods have experienced an increasing popularity for a large range of problems, including the solution of indefinite problems (see, e.g. [31,34] in the context of constrained optimization problems and fluid-structure interaction, respectively).

To define a MG method, we first introduce some notation. Let $\{\mathcal{A}_k\}_{k=1}^L$ be the linear system matrices representing the discretization of (1.4) on a sequence of L meshes, \mathcal{A}_{k-1} being on a coarser mesh than \mathcal{A}_k . Let further $\{P_k\}_{k=1}^{L-1}$ be the natural interpolation matrices relating variationally the system matrix \mathcal{A}_k to its coarser counterpart \mathcal{A}_{k-1} , and let \mathbf{y} be the generic input vector, \mathbf{x} the corresponding output vector and σ, ν, ℓ and tol be given constants, where ℓ represents the current level and tol the given tolerance. In Algorithm 3.1, we introduce our recursive preconditioner of nonlinear AMLI type:

Algorithm 3.1 The Multilevel Algorithm

```

MLPrecond( $\mathbf{x}, \mathbf{y}, \{\mathcal{A}_k\}_k, \{P_k\}_k, \sigma, \nu, \ell, tol$ ):
smooth  $\nu$  times on  $\mathcal{A}_\ell \mathbf{x} = \mathbf{y}$ ;
restrict residual:  $\mathbf{r} = P_{\ell-1}^T(\mathbf{y} - \mathcal{A}_\ell \mathbf{x})$ ;
if  $\ell - 1 = 1$  then
     $\mathbf{x}_c = (\mathcal{A}_{\ell-1}) \backslash \mathbf{r}$ ;    // coarsest level: Matlab notation for a direct method
else
     $\mathbf{x}_c = \mathbf{0}$ ;
     $Precond = @MLPrecond(\mathbf{x}_c, \mathbf{r}, \{A_k\}_{k=1}^{\ell-1}, \{P_k\}_{k=1}^{\ell-1}, \sigma, \nu, \ell - 1, tol)$ ;
    FGMRES( $\mathcal{A}_{\ell-1}, \mathbf{x}_c, \mathbf{r}, tol, \sigma, Precond$ );
end if
update  $\mathbf{x} = \mathbf{x} + P_{\ell-1} \mathbf{x}_c$ ;
smooth  $\nu$  times on  $\mathcal{A}_\ell \mathbf{x} = \mathbf{y}$ ;

```

Algorithm 3.1 is a multigrid cycle with $\sigma \geq 1$ recursive calls to coarse levels, which for $\sigma = 2$, is a variant of W-cycle MG; the method “MLPrecond” recursively calls itself σ times as a preconditioner inside an FGMRES scheme [28]). Since our matrices A_ℓ are indefinite, we use for smoothing a preconditioned Krylov subspace method, which also contributes towards the nonlinear nature of the proposed preconditioner.

3.2.1. Interpolation and Restriction. Starting with a mesh that is sufficiently coarse to allow a fast solution of the discrete system (e.g. with a direct solver that we have denoted with “ \backslash ”), we refine the mesh uniformly L times. With each mesh, we associate a corresponding triple of finite element spaces, $\mathbf{V}_k, Q_k, \mathcal{Z}_k$, $k = 1, \dots, L$. By construction, the coarse level spaces are subspaces of the next fine level spaces. This defines natural embeddings $\{P_k^u\}_{k=1}^L$, $\{P_k^p\}_{k=1}^L$ and $\{P_k^W\}_{k=1}^L$ which transfer (interpolate) the degrees of freedom of \mathbf{u} , p and W , respectively from the coarse mesh (level $k-1$) to the fine mesh (level k). The (monolithic) interpolation operator is then given by

$$P_k = \begin{bmatrix} P_k^u & 0 & 0 \\ 0 & P_k^p & 0 \\ 0 & 0 & P_k^W \end{bmatrix}.$$

The matrix \mathcal{A}_L is assembled on the finest mesh, and the coarse ones are variationally related via the Galerkin condition $\mathcal{A}_{k-1} = P_k^T \mathcal{A}_k P_k$ for $k = 1, \dots, L$.

3.2.2. Smoothing. Several types of smoothers may be considered. A classical approach is to perform a few iterations of a preconditioned iterative method with a simple (e.g., diagonal or Gauss–Seidel) preconditioner. Typically, in the case of s.p.d. problems, stationary iterations are the method of choice.

We will follow a similar idea on the indefinite system (2.3), which we precondition by an overlapping additive Schwarz method. For the sake of simplicity, we will omit the index k indicating the level of discretization for the remainder of this section. Given the discretized domain Ω on any given level, we may subdivide the domain into m overlapping subsets $\{\Omega_i\}_{i=1}^m$. Then we set up linear mappings $\{I_i^u\}_{i=1}^m$, $\{I_i^p\}_{i=1}^m$ and $\{I_i^W\}_{i=1}^m$ restricting the degrees of freedom of \mathbf{u} , p and W ,

respectively, to the local domain Ω_i . The i th subdomain local matrix is then given by

$$\mathcal{A}_i = I_i \mathcal{A} I_i^T \text{ with } I_i = \begin{bmatrix} I_i^u & 0 & 0 \\ 0 & I_i^p & 0 \\ 0 & 0 & I_i^W \end{bmatrix}$$

and the inverse of the global matrix is approximated by the additive Schwarz preconditioner defined by the formula

$$\mathcal{A}^{-1} \approx \sum_{i=1}^m I_i^T \mathcal{A}_i^{-1} I_i.$$

The size of the subdomains should be chosen sufficiently small so that the inverse of the local matrices \mathcal{A}_i can be easily computed.

4. Numerical results.

4.1. Implementation. Unless stated otherwise, the computational domain is a unit square or a unit cube. We discretize in space with P2-P1 finite elements for velocity and pressure. The auxiliary variable W is discretized with P1 finite elements. We refer to the C++ finite element library MFEM [24]. For the coarsest grid we choose an $h = 1/4$ grid in the two dimensional case and an $h = 1/2$ grid in three dimensions. As a solver on the coarsest grid we use a direct solver within the C library SUITESPARSE. When solving the Stokes type equations, the matrix \mathcal{A} is symmetric (and so is the coarse level matrix \mathcal{A}_1). Since \mathcal{A} has the form (2.3), it can be shown [32] that there exists a factorization as $\mathcal{P}^T \mathcal{A}_1 \mathcal{P} = LDL^T$ for any permutation matrix \mathcal{P} . Here L is a lower triangular and D a diagonal matrix. We use the LDL package [13] for computing this factorization. For the Navier-Stokes type equations, the linear system is in general not symmetric and the coarse level system is factorized by $\mathcal{A}_1 = LU$ (with a lower triangular L and upper triangular matrix U) using the UMFPACK package [11, 12, 15, 16]). Before computing all factorizations we apply a fill-in reducing reordering provided by AMD [1, 2, 14]. To set up the smoother on each level (except for the coarsest), we first generate an adjacency matrix $S = [s_{ij}]$ (with $s_{ij} = 1$ if element i and j share a common face in three dimensions or a common edge in two dimensions and $s_{ij} = 0$ otherwise). We then apply a graph partitioner in METIS [23] on S . This procedure results in a partitioning of the mesh in which the overlap consists of one layer of elements at the interface. Extra layers of overlap may be included as well. The solves on each subdomain is again done by the direct solvers provided in SUITESPARSE. Table 4.1 shows the different meshes we use for our experiments and the number of multigrid levels used for each mesh. Also, the number of subdomains is shown. The number of overlapping nodes is specified as well. The number of subdomains on each level has a strong influence on the performance of our preconditioner. The trade-off is between the size of the local system (not too large) and the overall efficacy of the smoother. This is achieved by increasing the number of subdomains by a factor of 4 in 2D and a factor of 6 in 3D for each additional multigrid level, as shown in the table. The size of the discrete system is shown as well. To produce the results in the following subsections, we start the nonlinear Picard iterations with the initial guess $\mathbf{u} = \mathbf{u}^0$, $p \equiv 0$, $W \equiv 0$ where \mathbf{u}^0 is the solution of $-\mu \Delta \mathbf{u}^0 = \mathbf{f}$ solved with preconditioned CG iterations. We continue the nonlinear iterations until

$$\frac{\|\mathbf{r}\|_2}{\|\mathbf{r}_0\|_2} \leq 10^{-2}$$

Experiments on Unit Square					
mesh	# levels	# subd.	#overlap. nodes	size overl.	size lin. syst.
$h = 1/8$	2	3	34-40	27	902
$h = 1/16$	3	9	42-50	126	3,334
$h = 1/32$	4	27	56-72	498	12,806
$h = 1/64$	5	81	68-85	1,839	50,182
$h = 1/128$	6	243	89-110	6,751	198,662
$h = 1/256$	7	729	114-143	24,343	790,534

Experiments on Unit Cube					
mesh	# levels	# subd.	size subd.	size overl.	size lin. syst.
$h = 1/4$	2	12	24-31	97	3,062
$h = 1/8$	3	72	27-49	669	19,842
$h = 1/16$	4	432	34-55	4,697	142,202
$h = 1/32$	5	2,392	41-70	34,925	1,075,434

TABLE 4.1

Number of levels of multigrid, number of subdomains, size of each subdomain, size of overlap and the size of the linear system to be solved in two and three dimensions.

where $\mathbf{r}(\mathbf{r}_0)$ is the current (initial) residual. We set the absolute tolerance to $5 \cdot 10^{-6}$. By choosing this nonlinear stopping criterion we make sure that the linear solver is accurate enough to achieve nonlinear convergence. The linear system is solved by **FGMRES** with our geometric nonlinear **AMLI** multigrid preconditioner and is considered converged if the quotient of current and initial residual drops below 10^{-6} in the L^2 -norm. All tables display the number of linear iterations needed for convergence of the first nonlinear iteration.

4.2. Choosing the regularization parameter. In Section 3.1 we stated that the performance of the preconditioner \mathcal{A}_ε improves as ε decreases, provided that the inverse $\mathcal{A}_\varepsilon^{-1}$ is computed exactly. However, the reduction of the regularization parameter in general deteriorates the conditioning properties of the matrix and this may impair the quality of the approximation. In this respect, finding the optimal value of ε involves finding the right trade-off between numerical stability and approximating the physical problem to be solved. In our experiments we empirically found that the optimal choice to be $\varepsilon = 10^{-2}$. However, a rigorous analysis justifying this choice is still missing. It is worth noticing that the domain decomposition used in our experiments is based entirely on the mesh and not on the solution. If a subdomain is entirely contained in a plug region, we may experience some performance degradation. As a matter of fact, the local representation of the linear system is extremely ill-conditioned for small values of ε and the local (direct) solves may be very inaccurate, resulting in failure of the smoother. Larger values of ε yield an improved conditioning of the local system and local solves are more accurate.

We also noticed that the condition number of the regularized block N_ε in (2.3) grows mildly as $\varepsilon \rightarrow 0$ except when between 10^{-2} and 10^{-3} where the increase is more evident (see Figure 4.1).

4.3. Flow between two parallel plates. This test case is one of the few examples in which the analytical solution is known for the steady (Navier) Stokes type Bingham problem. It describes

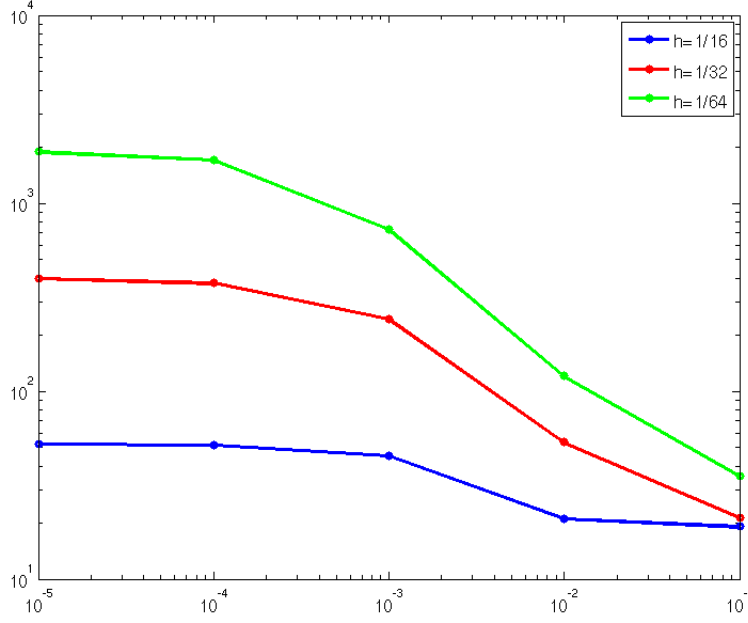


FIG. 4.1. Condition number of the block N_ϵ for different values of ϵ and h .

a flow between two parallel plates and its solution is given by

$$(4.1) \quad u_1 = \begin{cases} \frac{1}{8}[(1 - 2\tau_s)^2 + (1 - 2\tau_s - 2y)^2] & \text{if } 0 \leq y < \frac{1}{2} - \tau_s, \\ \frac{1}{8}(1 - 2\tau_s)^2, & \text{if } \frac{1}{2} - \tau_s \leq y \leq \frac{1}{2} + \tau_s, \\ \frac{1}{8}[(1 - 2\tau_s)^2 - (2y - 2\tau_s - 1)^2] & \text{if } \frac{1}{2} + \tau_s < y \leq 1, \end{cases}$$

$u_2 \equiv u_3 \equiv 0$ and $p = -x$. The strain rate vanishes in the plug region $\{(x, y, y) | \frac{1}{2} - \tau_s \leq y \leq \frac{1}{2} + \tau_s\}$. In our experiment we impose Dirichlet boundary conditions on the unit square and cube according to (4.1) with $\tau_s = 0.3$ and $\mu = 1$. To precondition the flexible **GMRES** iterations we use the algorithm from Section 3 with two smoothing steps ($\nu = 2$) in 2D and four smoothings ($\nu = 4$) in 3D as well as two iterations of **FGMRES** on each multigrid level ($\sigma = 2$). Table 4.2 displays the number of flexible **GMRES** iterations needed for convergence for the first Picard step, the total number of nonlinear iterations needed for convergence as well as the CPU time needed for solving the linearized system. In the three dimensional case the number of linear iterations slightly increases with the size of the mesh. However, the parameters for the preconditioner were chosen to minimize the CPU time as opposed to the iteration count. By properly tuning the number of smoothings or the number of subdomains we get mesh independence. Note that for the Stokes-type problem only the matrix N needs to be updated before each nonlinear iteration. In this respect, the timings provided in Table 4.2 for setting up the preconditioner are divided into initial setup time (this includes setting up the interpolations between the different levels, determining the subdivision of the domains and setting up the restriction operators for each subdomain) and updating time (this

Two Dimensional Experiments					
mesh	# lin. its	CPU time (s)	setup (s)	updating (s)	# nonlin. its.
$h = 1/8$	10	0.01	0.01	0.01	6
$h = 1/16$	13	0.08	0.02	0.03	6
$h = 1/32$	14	0.50	0.05	0.09	6
$h = 1/64$	14	2.19	0.18	0.40	6
$h = 1/128$	14	9.60	0.86	1.72	7
$h = 1/256$	12	37.93	5.44	7.25	7

Three Dimensional Experiments					
mesh	# lin. its	CPU time (s)	setup (s)	updating (s)	# nonlin. its.
$h = 1/4$	6	0.08	0.01	0.07	4
$h = 1/8$	8	1.51	0.17	0.75	5
$h = 1/16$	16	27.89	1.81	6.74	6
$h = 1/32$	11	258.53	25.53	90.46	6

TABLE 4.2

The flow between two parallel plates, an analytical test case in two and three dimensions: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of nonlinear (Picard) iterations.

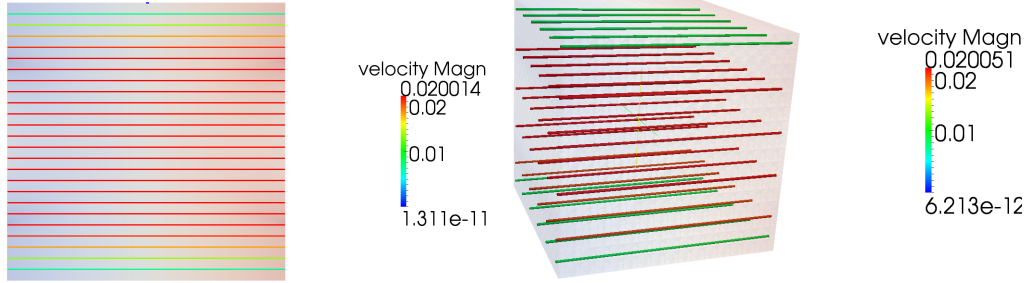


FIG. 4.2. Streamlines and pressure for the two and three dimensional flow between two parallel plates. The pressure field is indicated by the background color, the streamlines are colored by the magnitude of the velocity field.

includes the updating of N in the preconditioner, computing a factorization of the local matrices on each subdomain and computing a factorization for the direct solve on the coarsest level). All experiments are implemented in serial code. Timings for the Stokes-type Bingham problem are obtained on a personal laptop with an Intel Core i7 processor, 2.6 GHz and 8 GB of memory. Due to a higher memory requirement, experiments involving the Navier-Stokes type Bingham problems are run on a Sun Microsystems SunFire X4600, with 20 AMD Opteron(tm) cores and 32 GB of memory.

Figure 4.2 shows the streamlines and pressure of this flow in two and three dimensions.

Two Dimensional Experiments					
mesh	# lin. its	CPU time (s)	setup (s)	updating (s)	# nonlin. its.
$h = 1/8$	8	0.01	0.01	0.01	4
$h = 1/16$	11	0.07	0.01	0.02	5
$h = 1/32$	12	0.42	0.04	0.10	5
$h = 1/64$	12	1.87	0.17	0.40	5
$h = 1/128$	12	8.22	0.80	1.71	5
$h = 1/256$	11	34.91	5.44	7.29	4

Three Dimensional Experiments					
mesh	# lin. its	CPU time (s)	setup (s)	updating (s)	# nonlin. its.
$h = 1/4$	3	0.04	0.01	0.06	3
$h = 1/8$	5	0.94	0.17	0.74	4
$h = 1/16$	8	13.94	1.80	6.63	5
$h = 1/32$	5	167.21	25.48	89.12	5

TABLE 4.3

The lid-driven cavity flow in two and three dimensions (Stokes): Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of nonlinear (Picard) iterations.

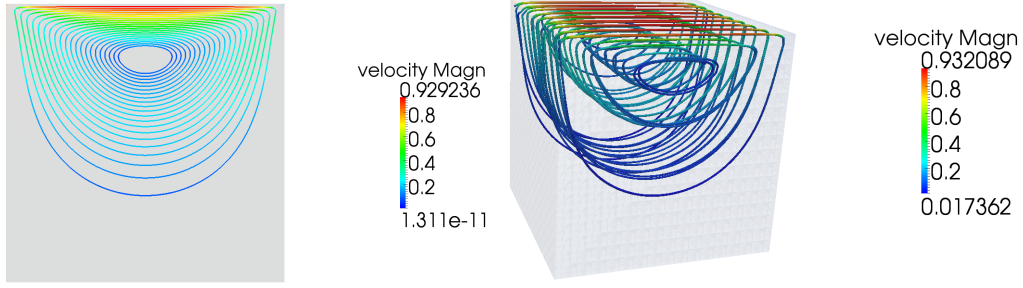


FIG. 4.3. Streamlines and pressure field of the lid-driven cavity (Stokes) in two (left) and three (right) dimensions.

4.4. The lid-driven cavity. This is a standard benchmark problem for CFD codes. The lid is moved at a velocity of magnitude 1 in the x -direction, i.e.

$$\mathbf{u} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ if } y = 1$$

and we impose homogeneous Dirichlet boundary conditions everywhere else. Again we set $\tau_s = 2$ and $\mu = 1$. For preconditioning we use the multilevel algorithm with two smoothings in 2D and four smoothings in 3D as well as two inner **GMRES** iterations on each level. Table 4.3 shows the numerical results for this experiment. Streamlines and pressure distribution of the lid-driven cavity are shown in Figure 4.3.

4.5. The steady Navier-Stokes type problem. We now apply the lid-driven cavity test case to the steady Navier-Stokes type problem. Here all specifications are the same as in the

Two Dimensional Experiments					
mesh	# lin. its	CPU time (s)	setup (s)	updating (s)	# nonlin. its.
$h = 1/8$	7	0.03	0.01	0.04	9
$h = 1/16$	10	0.41	0.01	0.19	8
$h = 1/32$	12	2.50	0.04	0.77	7
$h = 1/64$	12	11.74	0.23	3.27	6
$h = 1/128$	12	54.00	1.44	13.39	5
$h = 1/256$	11	232.63	11.87	53.32	4

Three Dimensional Experiments					
mesh	# lin. its	CPU time (s)	setup (s)	updating (s)	# nonlin. its.
$h = 1/4$	6	0.59	0.02	0.61	12
$h = 1/8$	6	9.28	0.12	7.33	17
$h = 1/16$	8	101.76	1.77	58.29	15
$h = 1/32$	7	747.84	47.53	440.48	4

TABLE 4.4

The lid-driven cavity flow for the Navier-Stokes type problem: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of Picard iterations.

previous subsection except that we now solve (1.4) with $\frac{\partial \mathbf{u}}{\partial t} \equiv 0$ and $\rho = 1$ and then we impose

$$\mathbf{u} = \begin{bmatrix} 50 \\ 0 \\ 0 \end{bmatrix} \quad \text{if } y = 1$$

corresponding to a Reynolds number $Re = 50$. We now tighten the nonlinear tolerance to 10^{-4} to ensure accurate solutions. Numerical results are shown in Table 4.4. Note the drop in the nonlinear iteration count for $h = 1/32$ in the three dimensional case. This is due to the higher Reynolds number in this test case; the relatively high number of nonlinear iterations for the coarser meshes are due to slight instabilities of the Picard iterations (the residual reduction was not monotone). We did not see this effect for lower Reynolds numbers ($Re \approx 10$), the effect was even more evident if we increased the Reynolds number (to approximately $Re = 50$). Streamlines and pressure of this flow are shown in Figure 4.4.

4.6. The unsteady Navier-Stokes type problem on a non-trivial geometry. This experiment is performed on a cylindrical domain with a sphere attached to it. It is meant as an idealized geometry that approximates a blood vessel with an aneurysm. This experiment serves as a first step in understanding the relevance of Bingham fluids in problems occurring in hemodynamics, see [21, 27]. We discretize the domain with second order isoparametric elements. Using elements of higher order has the effect that the “curved” shape of the domain is captured well during the refinement process of the geometric multigrid. See Figure 4.5 for the shape of the geometry on each multigrid level.

We use multigrid preconditioning on three levels in this experiment with four smoothings and two inner FGMRES iterations. In the Bingham fluid equations we take $\mu = 1$ and $\tau_s = 1$. We prescribe parabolic boundary conditions on the inflow, a no-slip condition on the walls and homogeneous Neumann boundary conditions on the outflow. Time is discretized on the interval from $t = 0$ to

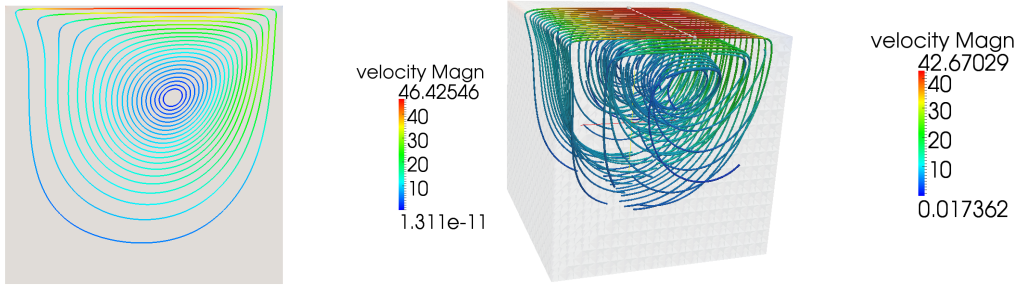


FIG. 4.4. Velocity and pressure field of the lid-driven cavity on the Navier-Stokes type problem.

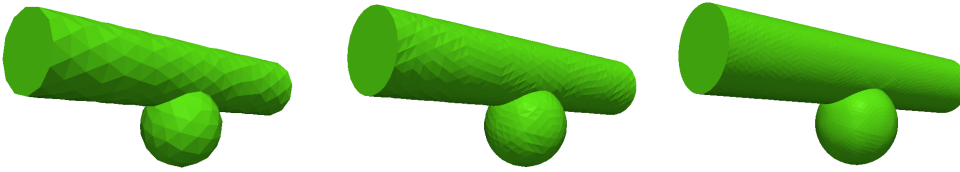


FIG. 4.5. Idealized blood vessel with aneurysm on three geometric multigrid levels. Left: The coarsest level (280 elements), center: one level of refinement (2,240 elements), right: two levels of refinement (17,920 elements).

$t = 1.5$ and a time step of $\Delta t = 0.1$. Table 4.5 shows more specifications on the preconditioner as well as the numerical results for this experiment. Figure 4.6 shows the streamlines and pressure of this flow after a steady state has been reached.

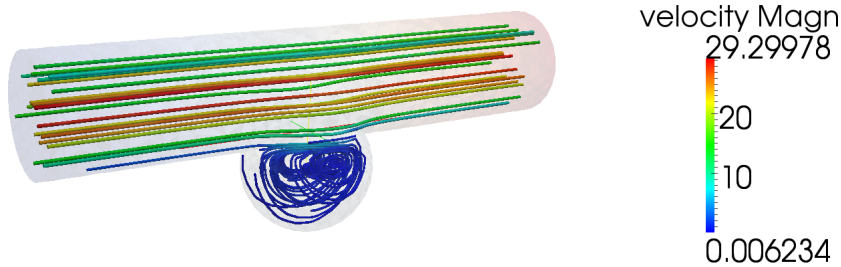


FIG. 4.6. Streamlines and pressure field of the unsteady Navier-Stokes type problem in a cylindrical domain with an attached sphere.

5. Conclusion. In this paper we have proposed a new way for solving the non-regularized Bingham fluid flow equations efficiently. We discretize and linearize the Bingham fluid flow in the mixed formulation proposed in [3] and solve the resulting linear systems by a flexible Krylov subspace method. Convergence of this iterative method is accelerated by a nonlinear AMLI (geometric) multigrid algorithm. The preconditioner is computed for the regularized problem and applied to

Fine level		Intermediate level		General Information	
# subd.:	72	# subd.:	12	# Picard its (per time st.):	5
size subd.:	100-140	size subd.:	74-98	CPU (s) (per t.s.):	53.41
size overl.:	2,791	size overl.:	342	setup (s):	0.80
				updating:	72.36
				# linear its:	6

TABLE 4.5

Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment.

the solution of the non-regularized Bingham system. This utilization of the parameter is novel in the sense that it serves as a preconditioning parameter for solving the non-regularized Bingham model as opposed to a parameter for regularization purposes. Our numerical experiments indicate mesh independent convergence in a low number of iterations, a rigorous proof of this being left for future research. Timings are obtained here on serial machines, but may be significantly improved on a parallel architecture. The application of our method to large-scale problems on parallel architectures, with particular reference to problems in computational hemodynamics is a natural next step of the present research.

REFERENCES

- [1] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 886–905.
- [2] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *Algorithm 837: AMD, an approximate minimum degree ordering algorithm*, ACM Trans. Math. Softw., 30 (2004), pp. 381–388.
- [3] A. APOSPORIDIS, E. HABER, M. A. OLSHANSKII, AND A. VENEZIANI, *A mixed formulation of the Bingham fluid flow problem: Analysis and numerical solution*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 2434 – 2446.
- [4] M. BENZI AND X.-P. GUO, *A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations*, Appl. Numer. Math., 61 (2011), pp. 66–76.
- [5] M. BENZI, M. NG, Q. NIU, AND Z. WANG, *A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations*, J. Comput. Phys., 230 (2011), pp. 6185–6202.
- [6] M. BENZI AND M. A. OLSHANSKII, *An Augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Sci. Comput., 28 (2006), pp. 2095–2113.
- [7] M. BENZI, M. A. OLSHANSKII, AND Z. WANG, *Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations*, International Journal for Numerical Methods in Fluids, 66 (2011), pp. 486–508.
- [8] M. BERCOVIER AND M. ENGELMAN, *A finite-element method for incompressible non-Newtonian flows*, Journal of Computational Physics, 36 (1980), pp. 313 – 326.
- [9] R. BYRON-BIRD, G. DAI, AND B. J. YARUSSO, *Rheology and flow of viscoplastic materials*, Reviews in Chemical Engineering, 1 (1983), pp. 1 – 70.
- [10] T. F. CHAN, G. H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1977.
- [11] T. A. DAVIS, *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Softw., 30 (2004), pp. 196–199.
- [12] T. A. DAVIS, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Trans. Math. Softw., 30 (2004), pp. 165–195.
- [13] T. A. DAVIS, *Algorithm 849: A concise sparse Cholesky factorization package*, ACM Trans. Math. Softw., 31 (2005), pp. 587–591.
- [14] T. A. DAVIS, *Direct Methods for Sparse Linear Systems*, Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, 2006.
- [15] T. A. DAVIS AND I. S. DUFF, *An unsymmetric-pattern multifrontal method for sparse lu factorization*, SIAM

- J. Matrix Anal. Appl., 18 (1997), pp. 140–158.
- [16] T. A. DAVIS AND I. S. DUFF, *A combined unifrontal/multifrontal method for unsymmetric sparse matrices*, ACM Trans. Math. Softw., 25 (1999), pp. 1–20.
 - [17] E. J. DEAN, R. GLOWINSKI, AND G. GUIDOBONI, *On the numerical simulation of Bingham visco-plastic flow: Old and new results*, Journal of Non-Newtonian Fluid Mechanics, 142 (2007), pp. 36–62.
 - [18] G. DUVAUT AND J. L. LIONS, *Inequalities in Mathematics and Physics*, Springer, 1998.
 - [19] H. C. ELMAN, D. J. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford Series in Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2005.
 - [20] H. C. ELMAN AND R. S. TUMINARO, *Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations*, Electr. Trans. Numer. Anal., 35 (2009).
 - [21] L. FORMAGGIA, A. QUARTERONI, AND A. VENEZIANI, *Cardiovascular Mathematics*, Springer, 2009.
 - [22] P. P. GRINEVICH AND M. A. OLSHANSKII, *An iterative method for the Stokes-type problem with variable viscosity*, SIAM J. Sci. Comput., 31 (2009), pp. 3959–3978.
 - [23] G. KARYPIS AND V. KUMAR, *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*, 2009.
 - [24] MFEM, *Finite element discretization library*. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, <http://code.google.com/p/mfem/>, 2012.
 - [25] E. A. MURAVLEVA AND M. A. OLSHANSKII, *Two finite-difference schemes for the calculation of Bingham fluid flows in a cavity*, Russ. J. Numer. Anal. Math. Model., 23 (2008), pp. 615–634.
 - [26] T. C. PAPANASTASIOU, *Flow of materials with yield*, J. Rheol., 31 (1987), pp. 385–404.
 - [27] A. M. ROBERTSON, A. SEQUEIRA, AND V. KAMENEVA, *Hemorheology in Hemodynamical Flows*, Birkhauser, Basel, 2008.
 - [28] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
 - [29] M. SAHIN AND H. J. WILSON, *A semi-staggered dilation-free finite volume method for the numerical solution of viscoelastic fluid flows on all-hexahedral elements*, Journal of Non-Newtonian Fluid Mechanics, 147 (2007), pp. 79 – 91.
 - [30] P. SARAMITO AND N. ROQUET, *An adaptive finite element method for viscoplastic fluid flows in pipes*, Comput. Meth. Appl. Mech. Eng., 190 (2001), pp. 5391–5412.
 - [31] J. SCHÖBERL, R. SIMON, AND W. ZULEHNER, *A robust multigrid method for elliptic optimal control problems*, SIAM J. Numer. Anal., 49 (2011), pp. 1482–1503.
 - [32] R. J. VANDERBEI, *Symmetric quasidefinite matrices*, SIAM Journal on Optimization, 5 (1995), pp. 100–113.
 - [33] P. S. VASSILEVSKI, *Multilevel Block Factorization Preconditioners*, Springer, 2008.
 - [34] H. YANG AND W. ZULEHNER, *Numerical simulation of fluid-structure interaction problems on hybrid meshes with algebraic multigrid methods*, J. Comput. Appl. Math., 235 (2011), pp. 5367–5379.